

# Jetson Nano

## 1. 刷机以及迁移系统到 SD 卡

刷机：[参考链接](#)

迁移到 SD 卡：[参考链接](#)

### 安装系统

1. ubuntu虚拟机或者主机打开终端，新建文件夹

```
sudo mkdir sources_nano
cd sources_nano
```

2. 下载路径

```
https://developer.nvidia.com/embedded/14t/r32_release_v7.2/t210/jetson-210_linux_r32.7.2_aarch64.tbz2
https://developer.nvidia.com/embedded/14t/r32_release_v7.2/t210/tegra_linux_sample-root-filesystem_r32.7.2_aarch64.tbz2
```

将资源包移到文件夹，并解压(实际操作的时候，请尽量用tab按键自动补齐指令)

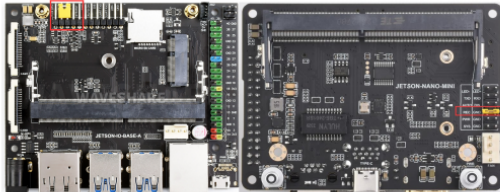
```
sudo mv ~/Downloads/Jetson-210_Linux_R32.7.2_aarch64.tbz2 ~/sources_nano/
sudo mv ~/Downloads/Tegra_Linux_Sample-Root-Filesystem-R32.7.2_aarch64.tbz2 ~/sources_nano/
```

3. 解压资源

```
sudo tar -xjf Jetson-210_Linux_R32.7.2_aarch64.tbz2
cd Linux_for_Tegra/rootfs/
sudo tar -xjf ../../Tegra_Linux_Sample-Root-Filesystem_R32.7.2_aarch64.tbz2
cd ../
sudo ./apply_binaries.sh (若出现报错按提示操作，再重新输入该指令)
```

### 硬件配置(进入recovery 模式)

1. 用跳帽或者杜邦线短接FC REC和GND引脚，位置如下图，位于核心板底下。
2. 连接DC电源到圆形供电口，稍等片刻。
3. 用USB线（注意要是数据线）连接Jetson Nano的Micro USB接口到Ubuntu主机。



### 系统烧录

1. 烧录系统，Jetson Nano 需进入recovery模式，连接到Ubuntu电脑。

```
cd ~/sources_nano/Linux_for_Tegra
sudo ./flash.sh jetson-nano-emmc mmcblk0p1
```

2. 烧录完成之后，去掉底板的跳帽，接入显示器，重新上电，按照提示进行开机配置（如果是设置的pre-config，上电后直接进入系统）。

经过测试，**完全按照教程步骤**就可以顺利安装系统。如果有迁移到 SD 卡的需求，推荐使用

“方法二：直接下载资源包”，使用 SDK Manager 安装后在后续步骤中可能会有问题。

## 2. Cuda 安装

根据上边教程安装的系统为纯净系统, 不包含 cuda 等必须依赖, 如果只需要基于 torch 运行 yolo 模型, 则安装 cuda 即可。

### 1. 更新 apt 源 [参考链接](#)

传统格式 ( `/etc/apt/sources.list` )

Ubuntu 版本: **Ubuntu 18.04 LTS (bionic)**

☐ 启用源代码

☐ 启用 proposed

☐ 强制安全更新使用镜像

```
# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic-backports main restricted universe multiverse

# 以下安全更新软件源包含了官方源与镜像站配置，如有需要可自行修改注释切换
deb http://ports.ubuntu.com/ubuntu-ports/ bionic-security main restricted universe multiverse
# deb-src http://ports.ubuntu.com/ubuntu-ports/ bionic-security main restricted universe multiverse

# 预发布软件源，不建议启用
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic-proposed main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu-ports/ bionic-proposed main restricted universe multiverse
```

先更换成清华源 注意切换系统版本

`sudo vi /etc/apt/sources.list` 删除原有内容替换成上方地址

然后执行 `apt-get update`

安装 cuda 执行 `sudo apt-get install cuda-toolkit-10-2`

完成后 将下列内容复制到 `~/.bashrc` 文件末尾

```
sudo vi ~/.bashrc
```

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-10.2/lib64
```

```
export PATH=$PATH:/usr/local/cuda-10.2/bin
```

```
export CUDA_HOME=$CUDA_HOME:/usr/local/cuda-10.2
```

执行 `source ~/.bashrc`

通过 `nvcc -V` 检验是否安装成功

### 3. Torch 安装

直接 pip 会找不到适配 aarch64 的版本

因此需要按照教程安装 NVIDIA 官方编译的版本 [参考链接](#)

#### Pytorch安装

1. 首先登录英伟达官网下载Pytorch安装包，我们以PyTorch v1.9.0为例

▼ PyTorch v1.9.0

- JetPack 4.4 [L4T R32.4.3] / JetPack 4.4.1 [L4T R32.4.4] / JetPack 4.5 [L4T R32.5.0] / JetPack 4.5.1 [L4T R32.5.1] / JetPack 4.6 [L4T R32.6.1]
  - Python 3.6 - torch-1.9.0-cp36-cp36m-linux\_aarch64.whl 5.0G

2. 下载依赖库

```
sudo apt-get install libjpeg-dev zlib1g-dev libpython3-dev libavcodec-dev libavformat-dev libswscale-dev libopenblas-base libopenmpi-dev
```

3. 安装Pytorch

```
sudo pip3 install torch-1.9.0-cp36-cp36m-linux_aarch64.whl
```

4. 验证Pytorch是否安装成功

```
python3
import torch
x = torch.rand(5, 3)
print(x)
```

torchvision 没有 whl 包 需要手动编译 同样按照教程即可安装

#### Torchvision安装

1. Torchvision版本要与Pytorch版本相匹配,我们前面安装的Pytorch版本为1.9.0, Torchvision安装 v0.10.0版本

##### Package Versions

Depending on your version of JetPack-L4T, different tags of the l4t-pytorch container are available, each with support for Python 3.6. Be sure to clone a tag that matches the version of JetPack-L4T that you have installed on your Jetson.

- JetPack 4.6 (L4T R32.6.1)
  - l4t-pytorch:r32.6.1-pt1.9-py3
    - PyTorch v1.9.0
    - torchvision v0.10.0
    - torchaudio v0.9.0
  - l4t-pytorch:r32.6.1-pt1.8-py3
    - PyTorch v1.8.0
    - torchvision v0.9.0
    - torchaudio v0.8.0

### 4. 运行 YOLO

1. 最新版本的 yolo v5 ([github.com/ultralytics/yolov5.git](https://github.com/ultralytics/yolov5.git))

会因为 gitpython 不匹配 python 3.6 的原因无法安装 可以选择从 requirements.txt 中

去除或使用旧版:

链接: <https://pan.baidu.com/share/init?surl=USNqOdzgiHLbfkiaXonjog>

提取码: oq81

2. 在安装 matplotlib 和 opencv 的过程中容易遇到编译问题 可以使用

[https://gitee.com/PolarisF/jetson\\_pkgs/blob/master/matplotlib-3.3.4-cp36-](https://gitee.com/PolarisF/jetson_pkgs/blob/master/matplotlib-3.3.4-cp36-)

[cp36m-linux\\_aarch64.whl](#)

opencv: [https://blog.csdn.net/weixin\\_43220532/article/details/109156164](https://blog.csdn.net/weixin_43220532/article/details/109156164)

### 3. 手动扩充虚拟内存可以避免读取权值环节卡死 [参考链接](#)

#### 2.2 修改Nano板的显存

##### 1.打开终端输入:

```
1 | sudo gedit /etc/systemd/nvzramconfig.sh
```

##### 2.修改nvzramconfig.sh文件:

在打开的文档中找到mem值, 修改mem的值, 如下:

```
1 | 找到的 mem = $(((("${totalmem}"/2/"${NRDEVICES}")*1024))
2 | 我们修改的 mem = $(((("${totalmem}"*2/"${NRDEVICES}")*1024))
```

把 / 修改为 \*

##### 3.重启:

```
1 | reboot
```

##### 4.终端输入:

```
1 | free -h
```

可查看到swap已经变为7.7G

### 5. Tensor RT 加速

<https://github.com/wang-xinyu/tensorrtx/tree/yolov5-v7.0/yolov5>

### 6. 各种可能需要的包适配 Jetson 平台的版本

<https://github.com/miooochi/jetson-packages-family>

[https://gitee.com/PolarisF/jetson\\_pkgs](https://gitee.com/PolarisF/jetson_pkgs)